

**PATENT****MS174293.01/MSFTP243US****CERTIFICATE OF FACSIMILE TRANSMISSION**

I hereby certify that this correspondence (along with any paper referred to as being attached or enclosed) is being faxed to 703-872-9306 on the date shown below to Mail Stop Appeal Brief Patents, Commissioner for Patents, P.O. Box 1450, Alexandria, Virginia 22313-1450.

**RECEIVED  
CENTRAL FAX CENTER**
**FEB 23 2005**Date: 2-23-05


Himanshu S. Amin

**IN THE UNITED STATES PATENT AND TRADEMARK OFFICE**

In re patent application of:

Applicant(s): Raja Krishnaswamy, *et al.*

Examiner: Nabil M. El Hady

Serial No: 09/893,829

Art Unit: 2154

Filing Date: June 28, 2001

Title: TRANSPARENT AND SUB-CLASSABLE PROXIES

**Mail Stop Appeal Brief – Patents  
Commissioner for Patents  
P.O. Box 1450  
Alexandria, VA 22313-1450**

**APPEAL BRIEF**

Dear Sir:

Applicant submits this brief in connection with an appeal of the above-identified patent application. A credit card payment form is filed concurrently herewith in connection with all fees due regarding this appeal brief. In the event any additional fees may be due and/or are not covered by the credit card, the Commissioner is authorized to charge such fees to Deposit Account No. 50-1063 [MSFTP243US].

02/24/2005 BBONNER 00000022 09893829

01 FC:1401	500.00	OP
02 FC:1251	120.00	OP

---

09/893,829

---

MS174293.01/MSFTP243US**I. Real Party in Interest (37 C.F.R. §41.37(c)(1)(i))**

The real party in interest in the present appeal is Microsoft Corporation, the assignee of the present application.

RECEIVED  
CENTRAL FAX CENTER  
FEB 23 2005

**II. Related Appeals and Interferences (37 C.F.R. §41.37(c)(1)(ii))**

Appellants, appellants' legal representative, and/or the assignee of the present application are not aware of any appeals or interferences which may be related to, will directly affect, or be directly affected by or have a bearing on the Board's decision in the pending appeal.

**III. Status of Claims (37 C.F.R. §41.37(c)(1)(iii))**

Claims 1-27 stand rejected by the Examiner. The rejection of claims 1-27 is being appealed.

**IV. Status of Amendments (37 C.F.R. §41.37(c)(1)(iv))**

No claim amendments have been entered after the Final Office Action.

**V. Summary of Claimed Subject Matter (37 C.F.R. §41.37(c)(1)(v))****A. Independent Claim 1**

Independent claim 1 recites a system for interacting with an object, the system comprising: a method call interceptor operable to intercept a method call to an object and to route the method call to a proxy, the method call interceptor accessible to application code; and an application code generic proxy operable to receive an intercepted method call, the application code generic proxy further operable to invoke a method on the object, to receive results from the object and to pass results to the entity that generated the intercepted method call. (See e.g., page 3, lines 10-27).

**B. Independent Claim 14**

Independent claim 14 recites a computer readable medium containing computer executable components for a system for interacting with an object, the components

---

09/893,829

---

MS174293.01/MSFTP243US

comprising: a method call intercepting component operable to intercept a method call to an object and to route the method call to a proxy, the method call intercepting component accessible to application code; and an application code generic proxy component operable to receive an intercepted method call, the application code generic proxy component further operable to invoke a method on the object, to receive results from the object and to pass results to the entity that generated the intercepted method call. (*See e.g., page 3, line 24-page 4, line 5.*)

**C. Independent Claim 15**

Independent claim 15 recites a method for interacting with an object, the method comprising: creating a base class proxy object; creating an application code generic proxy, where the application code generic proxy inherits from the base class proxy object; overriding a base class method defined in the base class, where the overridden method will receive an intercepted method call; intercepting a method call on the object; routing the method call to the application code generic proxy; invoking a method on the object; receiving a first result from the object; and returning a second result to the entity that generated the intercepted method call. (*See e.g., page 4, lines 6-26.*)

**D. Independent Claim 23**

Independent claim 23 recites a computer readable medium containing computer executable instructions for performing a method for interacting with an object, the method comprising: creating a base class proxy object; creating an application code generic proxy, where the application code generic proxy inherits from the base class proxy object; overriding a base class method defined in the base class, where the overridden method will receive an intercepted method call; intercepting a method call on the object; routing the method call to the application code generic proxy; invoking a method on the object; receiving a first result from the object; and returning a second result to the entity that generated the intercepted method call. (*See e.g., page 4, line 24-page 5, line 6.*)

09/893,829MS174293.01/MSFTP243US**E. Independent Claim 24**

Independent claim 24 recites a data packet adapted to be transmitted between two or more computer processes, the data packet comprising: one or more identifier/value pairs, the identifier identifying the value associated with the identifier/value pair, and the value providing information associated with an intercepted method call on an object. (See e.g., page 5, lines 7-20).

**F. Independent Claim 27**

Independent claim 27 recites a system for interacting with an object, the system comprising: means for creating a base class proxy object from application code, where the base class proxy object has a method that can be overridden by an inheriting application code generic proxy so that the overridden method can receive an intercepted method call. (See e.g., page 2, lines 15-20). Further independent claim 27 provides means for creating the application code generic proxy, where the application code generic proxy inherits from the base class proxy object and where the application code generic proxy overrides the base class method that can be overridden. (See e.g., page 2, line 21-page 3, line 9). Independent claim 27 also provides means for intercepting a method call and for transferring control to the overridden base class method in the application code generic proxy. (See e.g., page 2, lines 27-30). In addition, independent claim 27 provides means for the application code generic proxy to receive the intercepted method call. (See e.g., page 3, lines 1-4). Independent claim 27 also provides means for providing the overridden base class method with a call data structure associated with the intercepted method call. (See e.g., page 4, lines 9-12). In addition, independent claim 27 provides means for the application code generic proxy to invoke the method on the object, as well as means for the application code generic proxy to receive a first result from the object; and means for the application code generic proxy to return a second result to the entity that generated the intercepted method call. (See e.g., page 4, line 30-page 5, line 6).

The aforementioned means for limitations are identified as claim elements subject to the provisions of 35 U.S.C. §112 ¶6. The corresponding structures are identified with reference to the specification and drawings in the parentheticals above corresponding to

---

09/893,829

MS174293.01/MSFTP243US

those claim limitations.

**VI. Grounds of Rejection to be Reviewed (37 C.F.R. §41.37(c)(1)(vi))**

**A. Whether claims 1-27 are anticipated by Cohen *et al.* (US 6,324,543) under 35 U.S.C. §102(e).**

**VII. Argument (37 C.F.R. §41.37(c)(1)(vii))**

**A. Rejection of Claims 1-27 Under 35 U.S.C. §102(e)**

Claims 1-27 stand rejected under 35 U.S.C. §102(e) as being anticipated by Cohen *et al.* (US 6,324,543). This rejection should be withdrawn for at least the following reasons. Cohen *et al.* fails to disclose each and every limitation set forth in the subject claims.

A single prior art reference anticipates a patent claim only if it expressly or inherently describes *each and every limitation set forth in the patent claim*. *Trintec Industries, Inc. v. Top-U.S.A. Corp.*, 295 F.3d 1292, 63 USPQ2d 1597 (Fed. Cir. 2002); *See Verdegaal Bros. v. Union Oil Co. of California*, 814 F.2d 628, 631, 2 USPQ2d 1051, 1053 (Fed. Cir. 1987). The *identical invention must be shown in as complete detail as is contained in the ... claim*. *Richardson v. Suzuki Motor Co.*, 868 F.2d 1226, 9 USPQ2d 1913, 1920 (Fed. Cir. 1989) (emphasis added).

*i.* **Claims 1, 14, 15, 23 and 27**

Independent claims 1, 14, 15, 23 and 27 relate to a system and method for interacting with an object. The subject independent claims recite similar limitations, namely: a method call interceptor operable to intercept a method call to an object and to route the method call to a proxy, the method call interceptor accessible to application code; and an application code generic proxy operable to receive an intercepted method call, the application code generic proxy further operable to invoke a method on the object, to receive results from the object and to pass results to the entity that generated the intercepted method call. It is apparent that the invention as claimed

---

09/893,829

---

MS174293.01/MSFTP243US

utilizes a method call interceptor accessible to application code to intercept a method call to an object. Upon receipt of the method call by the method call interceptor, the method call interceptor routes the received method call to an application code generic proxy. The application code generic proxy upon receiving the intercepted method call is capable of invoking a method on the object, receiving the results of the invocation of the method on the object, and returning the results received from the object to the entity that generated the intercepted method call. Cohen *et al.* fails to teach or suggest these novel aspects of the claimed invention.

Cohen *et al.* discloses a method and system that allows a program to become dynamically reconfigurable without programmer intervention, i.e. the programs can be dynamically distributed among multiple computers within a computer network without modification to the source code of the programs running on the system. (See, Abstract). The Final Office Action (dated August 26, 2004) and the Advisory Action (dated November 15, 2004) suggest that Cohen *et al.* discloses *a method call interceptor operable to intercept a method call to an object and to route the method call to a proxy*, at col. 2, lines 1-5; and col. 3, lines 57-63, proxy B'. The noted passages however, disclose the generation of local and remote proxies by a Dynamic Object Distribution (DOD) system, wherein proxies intercept method calls to and from an object. It is apparent therefore that Cohen *et al.* discloses a proxy for intercepting and routing method calls to and from an object, rather than a *method call interceptor* for intercepting method calls and routing the intercepted method call to a proxy, whereupon the proxy directs the intercepted method call to the object. Further, it is apparent from Cohen *et al.* that it is the *proxy* that always intercepts method calls between the entity that generated the method call and the object that received the method call; similarly, it is the *proxy* that intermediates between the object that received the method call and the entity that generated the method call. Applicants' invention on the other hand, utilizes the method call interceptor only to intercept the method call from the entity that generated the method call; it is the application code generic proxy that passes the results generated by the object back to the entity that generated the method call. Thus, it is submitted, Cohen *et al.* fails to disclose or suggest the method call interceptor as claimed, and as a consequence applicants' invention is clearly distinguishable on this ground alone.

---

09/893,829

MS174293.01/MSFTP243US

Further, in the Final Office Action (dated August 26, 2004) it was asserted that Cohen *et al.* provided a *method call interceptor accessible to application code*, at col. 2, lines 1-10. In response, applicants' representative had contended that since Cohen *et al.* failed to provide a method call interceptor it was inconceivable as to how the cited document could possibly teach a method call interceptor accessible to application code. Nevertheless, despite Cohen *et al.*'s deficiency in this respect, it was posited that the cited passage merely provided a bytecode modification tool that modified bytecode files prior to the generation of proxies by the DOD system, i.e., the proxies once generated were not accessible to the bytecode modification tool, let alone the application code. Thus, it was submitted that applicants' claimed invention was distinguishable from Cohen *et al.* in this regard, as the invention as claimed provided a method call interceptor accessible to application code. The Advisory Action (dated November 15, 2004) fails however to address, let alone meet the full force of, the applicants' representative's assertion with respect to this particular shortcoming in Cohen *et al.*

Furthermore, the Examiner is reminded that the standard by which anticipation is to be measured is *strict identity* between the cited document and the invention as claimed, not mere equivalence or similarity. See, *Richardson* at 9 USPQ2d 1913, 1920. This means that in order to establish anticipation under 35 U.S.C. §102, the single document cited must not only expressly or inherently describe each and every limitation set forth in the patent claim, but also the identical invention must be shown in as complete detail as is contained in the claim. The fact that Cohen *et al.* fails to provide a method call interceptor, but rather a proxy that intercepts method calls, and further, in view of the fact that the proxies that are generated by Cohen *et al.* are not accessible to application code, leads one to believe that the cited document in the final analysis, does not provide an invention identical to that recited in the subject claims.

In view of at least the foregoing it is submitted that Cohen *et al.* fails to disclose all the limitations set forth in independent claims 1, 14, 15, 23 and 27, and associated dependent claims, and that this rejection should be withdrawn.

---

09/893,829MS174293.01/MSFTP243US

---

ii. Claim 24

Independent claim 24 recites a data packet adapted to be transmitted between two or more computer processes, wherein the data packet comprises: one or more identifier/value pairs, the identifier identifying the value associated with the identifier/value pair, and the value providing information associated with an intercepted method call on an object. In applicants' Reply to Final Office Action (dated August 26, 2004), it was argued in response to the Examiner's assertion that Cohen *et al.* inherently disclosed data packets comprising one or more identifier/value pairs providing information associated with an intercepted method call; that Cohen *et al.* was silent with regard to the utilization of identifier/value pairs, and that the supposed inherency upon which the Final Office Action sought to rely was manifestly absent. Further, the applicants' representative contended, and reiterates herein, that all that is apparent from Cohen *et al.*'s disclosure is the provision of a method that allows programs to become dynamically reconfigurable without programmer intervention, such that programmers can specify the conditions under which reconfiguration of programs can occur. The Advisory Action (dated November 15, 2004) once again fails to address, let alone meet the full force of, the applicants' representative's contentions with respect to these particular shortcomings in Cohen *et al.*.

Moreover, as stated *supra*, the Examiner is once again reminded that the standard by which anticipation is to be measured is *strict identity* between the cited document and the invention as claimed, not mere equivalence or similarity. This implies that in order to establish anticipation under 35 U.S.C. §102, the single document cited must not only expressly or inherently describe each and every limitation set forth in the patent claim, but also the identical invention must be shown in as complete detail as is contained in the claim. Since Cohen *et al.* is silent regarding the disclosure, let alone the utilization, of identifier/value pairs, is substantiation enough to deduce that Cohen *et al.* neither describes all the limitations set forth in the subject claim, nor provides an invention identical to that contained in the claim at issue. Accordingly, it is submitted that the Examiner has failed to meet the sufficiency of his burden under 35 U.S.C. §102, and that the rejection with respect to independent claim 24, and its associated dependent claims, should be withdrawn.

---

09/893,829

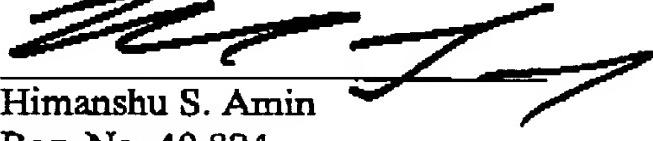
---

MS174293.01/MSFTP243US**B. Conclusion**

For at least the above reasons, the claims currently under consideration are believed to be patentable over the cited references. Accordingly, it is respectfully requested that the rejections of claims 1-27 be reversed.

If any additional fees are due in connection with this document, the Commissioner is authorized to charge those fees to Deposit Account No. 50-1063.

Respectfully submitted,  
AMIN & TUROCY, LLP



Himanshu S. Amin  
Reg. No. 40,894

AMIN & TUROCY, LLP  
24<sup>th</sup> Floor, National City Center  
1900 East 9<sup>th</sup> Street  
Telephone: (216) 696-8730  
Facsimile: (216) 696-8731

---

09/893,829

---

MS174293.01/MSFTP243US**VIII. Claims Appendix (37 C.F.R. §41.37(c)(1)(viii))**

1. A system for interacting with an object, the system comprising:
  - a method call interceptor operable to intercept a method call to an object and to route the method call to a proxy, the method call interceptor accessible to application code; and
    - an application code generic proxy operable to receive an intercepted method call, the application code generic proxy further operable to invoke a method on the object, to receive results from the object and to pass results to the entity that generated the intercepted method call.
2. The system of claim 1 where the object is located across a remoting boundary.
3. The system of claim 2 where the object is marshaled by reference.
4. The system of claim 2 where the object is marshaled by value.
5. The system of claim 1 where the method call interceptor is further operable to populate a call information data store with information associated with the intercepted method call, the call information data store accessible to the application code generic proxy.
6. The system of claim 5 where the call information data store is populated with at least one of a method name, one or more input parameters, a count of the number of input parameters, one or more type identifiers associated with the input parameters, a count of the number of return parameters for the method call, one or more type identifiers associated with the return parameters, class/interface defining method data, a stack pointer and a heap pointer.
7. The system of claim 6, where the call information data store is a message object that can be serialized and passed across a remoting boundary.

---

09/893,829

MS174293.01/MSFTP243US

8. The system of claim 1 where the method call interceptor is further operable to transfer control to a method in the application code generic proxy, where the method in the application code generic proxy overrides a base class method defined in a base class object from which the application code generic proxy inherits.
9. The system of claim 1 where the application code generic proxy is operable to perform proxy pre-processing before invoking the method on the object.
10. The system of claim 9 where the proxy pre-processing comprises at least one of load-balancing, transaction processing, object migration, object persisting, monitoring remote method calls, caching local data, caching remote data, controlling remote method call invocations and machine learning involved in optimizing remote method call invocation.
11. The system of claim 1 where the application code generic proxy is operable to perform proxy post-processing after receiving the results from the object.
12. The system of claim 11 where the proxy post-processing comprises at least one of auditing, transaction processing, object migration, object persisting, monitoring remote method calls, caching local data, caching remote data, controlling remote method call invocations and machine learning involved in optimizing remote method call invocation.
13. The system of claim 1 where the application code generic proxy invokes the method on the object by invoking a method available in a remoting infrastructure.
14. A computer readable medium containing computer executable components for a system for interacting with an object, the components comprising:

---

09/893,829

---

MS174293.01/MSFTP243US

a method call intercepting component operable to intercept a method call to an object and to route the method call to a proxy, the method call intercepting component accessible to application code; and

an application code generic proxy component operable to receive an intercepted method call, the application code generic proxy component further operable to invoke a method on the object, to receive results from the object and to pass results to the entity that generated the intercepted method call.

15. A method for interacting with an object, the method comprising:

creating a base class proxy object;

creating an application code generic proxy, where the application code generic proxy inherits from the base class proxy object;

overriding a base class method defined in the base class, where the overridden method will receive an intercepted method call;

intercepting a method call on the object;

routing the method call to the application code generic proxy;

invoking a method on the object;

receiving a first result from the object; and

returning a second result to the entity that generated the intercepted method call.

16. The method of claim 15 where the application code generic proxy performs proxy pre-processing before invoking the method on the object.

17. The method of claim 16 where the proxy pre-processing comprises at least one of load-balancing, transaction processing, object migration, object persisting, monitoring remote method calls, caching local data, caching remote data, controlling remote method call invocations and machine learning involved in optimizing remote method call invocation.

---

09/893,829

---

MS174293.01/MSFTP243US

18. The method of claim 15 where the application code generic proxy performs proxy post-processing before returning the result to the entity that generated the intercepted method call.
19. The method of claim 18 where the proxy post-processing comprises at least one of auditing, transaction processing, object migration, object persisting, monitoring remote method calls, caching local data, caching remote data, controlling remote method call invocations and machine learning involved in optimizing remote method call invocation.
20. The method of claim 15 where the object is located across a remoting boundary.
21. The method of claim 20 where the object is marshaled by reference.
22. The method of claim 20 where the object is marshaled by value.
23. A computer readable medium containing computer executable instructions for performing a method for interacting with an object, the method comprising:
  - creating a base class proxy object;
  - creating an application code generic proxy, where the application code generic proxy inherits from the base class proxy object;
  - overriding a base class method defined in the base class, where the overridden method will receive an intercepted method call;
  - intercepting a method call on the object;
  - routing the method call to the application code generic proxy;
  - invoking a method on the object;
  - receiving a first result from the object; and
  - returning a second result to the entity that generated the intercepted method call.

---

09/893,829

MS174293.01/MSFTP243US

24. A data packet adapted to be transmitted between two or more computer processes, the data packet comprising:

one or more identifier/value pairs, the identifier identifying the value associated with the identifier/value pair, and the value providing information associated with an intercepted method call on an object.

25. The data packet of claim 24 where the information associated with an intercepted method call on an object comprises at least one of method name, one or more input parameters, a count of the number of parameters input to the method, one or more type identifiers associated with the input parameters, a count of the number of return parameters for the method, one or more type identifiers associated with the return parameters, class/interface defining method data, a stack pointer and a heap pointer.

26. The data packet of claim 25, where the data packet is a serializable message object that can be passed across a remoting boundary.

27. A system for interacting with an object, the system comprising:

means for creating a base class proxy object from application code, where the base class proxy object has a method that can be overridden by an inheriting application code generic proxy so that the overridden method can receive an intercepted method call;

means for creating the application code generic proxy, where the application code generic proxy inherits from the base class proxy object and where the application code generic proxy overrides the base class method that can be overridden;

means for intercepting a method call and for transferring control to the overridden base class method in the application code generic proxy;

means for the application code generic proxy to receive the intercepted method call;

means for providing the overridden base class method with a call data structure associated with the intercepted method call;

09/893,829MS174293.01/MSFTP243US

means for the application code generic proxy to invoke the method on the object;

means for the application code generic proxy to receive a first result from the object; and

means for the application code generic proxy to return a second result to the entity that generated the intercepted method call.

**IX. Evidence Appendix (37 C.F.R. §41.37(c)(1)(ix))**

None.

**X. Related Proceedings Appendix (37 C.F.R. §41.37(c)(1)(x))**

None.